Overview of Applicants' claimed invention

The pending claims describe an improved caching mechanism which delivers a cached response to any incoming data request that is logically equivalent to a prior request, even though the prior request may have different character content. As claimed, applicants' invention first converts each incoming request message into a standard canonical form and then compares that canonical incoming message with previously received and stored canonical requests. If a match is found, a cached response data that was previously transmitted in response to the previously received matching request is returned.

Applicants' invention avoids the need to perform transaction processing to respond to a received request that is logically the same as but not identical to a prior request. Applicants' claimed invention allows a cached response to be returned whenever an incoming request is logically equivalent to a cached canonical request, even though its character content may differ from the original request. This in turn enables the system to immediately return cached responses without any additional processing, avoiding the need to access and package response data into a return message and avoiding the need to cache a further return message that duplicates a message that has already been cached.

The rejection of claim 1 in view of Fradette and Graham

As acknowledged by the Examiner, Fradette does not disclose converting the incoming request message into an incoming canonical request message expressed in a predetermined standard form as claimed.

Fradette describes a conventional caching mechanism that receives read and write requests along with the storage device addresses from which data is to be read or to which data is to be written. If the data requested in a read request is in the cache, the cached copy is supplied, otherwise the data stored at the specified device address is fetched.

Fradette also describes a system interface between the storage system that includes multiple storage devices (each of which may have its own cache) and a diverse collection of different client systems which issue I/O requests in different protocols. The system interface allows multiple storage devices to appear to represent a single virtual disk image in which addresses are represented as logical addresses. This interface "normalizes" I/O requests by

converting the logical address expressed in the host (client) I/O request into an internal storage address for use by the physical storage system. As the Examiner acknowledges, Fradette does not describe converting incoming request messages into incoming canonical request messages expressed in a predetermined standard form, and Fradette does not disclose comparing such incoming canonical request messages with previously received and stored canonical request messages.

The Examiner contends that it would have been obvious to modify the Fradette system to incorporate the teachings of Graham. Reconsideration is requested.

Graham describes a Universal Service Broker Interchange Mechanism (USBIM) which permits client processes that use many different protocols to obtain service advertisements from service providers that also use many different protocols. Each service provider employs a service protocol adapter servlet to translate its service advertisement a canonical form (preferably an XML document which conforms to a predetermined Document Type Definition) before it is posted into an advertisement registry. In this way, all of the service advertisements from all of the service providers, regardless of the protocol used by the individual service providers, are stored in a standard format in the advertisement registry. When a client issues a request for an advertisement using its native protocol, that request is converted into a standard canonical form that is compatible with the form in which the advertisements are stored.

But it would not have been obvious to one of ordinary skill in the art to modify the Fradette caching system in view of Graham to place incoming requests in canonical form as suggested by the Examiner. A person of ordinary skill in art would have no reason to attempt such a modification.

Before one skilled in the art can provide a solution to a problem, he or she must be aware of the problem, or be made aware of the problem and solution from some source. But Fradette nowhere indicates an awareness of the problem that a different request that is logically equivalent to a prior request will not produce a cached response. That follows from the fact that each cache store in Fradette's system is directly associated with a given storage device and compares the device data address in a read request with the storage addresses of data in the cache to determine whether or not requested data is in the cache. At this physical device address level, there is no concern that an incoming request might be logically equivalent to a different prior

address, so the problem encountered when handling, for example, complex XML HTTP requests which may be logically equivalent even though they are differently expressed does not arise.

Graham likewise does not suggest that problem or its solution. While Graham converts incoming request messages into canonical form, it does so for a completely different reason and in a way that is unlike the mechanism claimed by appellants. Graham converts requests using different protocols into standard canonical form so that service advertisements can be retrieved. Graham's canonicalization is not performed to cache data, the canonical requests are not stored as claimed, nor are canonical requests compared with prior requests. Because Graham's mechanism for performing canonical conversion is used in a totally different context, it could not be used to modify Fradette's caching system and, even there was some way in which such a modification could be made, it would not yield the claimed combination.

In short, if one skilled in the art considered Fradette and Graham, singly or in combination, they would find no disclosure whatsoever of either the problem solved by applicants' invention or the solution claimed. The person of ordinary skill would find not find any suggestion in either reference that prior art caching systems fail to properly handle requests that are logically equivalent but have different character content. Moreover, the system taught by Graham for permitting service advertisements to be retrieved when the service providers and clients use different protocols has nothing to do with caching and does not deal with either the problem or the solution.

The rejection of claims 3-5, 10 and 12-14 in view of Fradette, Graham and Mattis

Claims 3-5 are dependent on claim 1 and are allowable over the combination of Fradette and Graham for the reasons given above.

The Examiner acknowledges Fradette-Graham do not disclose generating and using an access key as set forth in claims 3-5, but suggests that feature is described in analogous art by Mattis, and that it would have been obvious to combine the teaching of Mattis "in order to provide an efficient method of comparing elements in the cache such as the one taught by of Fradette by generating an access key based on the request, thereby increasing efficiency of the cache mechanism, thereby resulting in increased throughput of the overall system."

Mattis describes a cache which compares incoming requests with prior requests and, if a

match is found, returns the response to the prior request to eliminate the need to again create and format a response. But Mattis does not handle the caching of logically equivalent but different requests as claimed by appellants and does not translate incoming messages into canonical request messages or compare them with prior messages. Because there is no canonical request message in Mattis, there is way to generate an access key based on the content of a canonical message as required by claims 3-5. Moreover, one skilled in the art would not modify the cache taught by Fradette to incorporate an access key lookup mechanism because Fradette's device cache uses physical storage addresses, and there would be no need to employ a complex key-based access mechanism of the type used by Mattis (in which keys are constructed by applying a hash function to the composition of the name or URL of the object and other information characterizing a request.)

Claims 10 and 12-14 were rejected for similar reasons with regard to claims 1 and 3-5 and are believed to be allowable for the reasons given above.

The rejection of claims 2, 6-9, 11, and 15-18 in view of Fradette, Graham and Schroeder

Applicants' claims 2, 6-9, 11 and 15-18 further specify that all or part of the request messages which are translated into canonical form are expressed in XML. As pointed out in applicants' specification, data requests expressed in XML may be logically identical but have different content; for example, logically identical XML request messages may have different line ending characters or include different whitespace characters which change the form but not the logical meaning of the request.

Applicants' claimed technique of converting incoming XML requests into canonical form for storage and comparison permits logically identical requests to be identified even though they don't have identical content as received. As explained at pages 7-8 of appellants' specification, 14 different conversion steps are performed in order to canonicalize an incoming XML document. Nothing in any of the cited references discloses or suggests canonicalizing XML request messages. As the Examiner acknowledges, neither Fradette nor Graham discloses that a portion of the incoming request message be expressed in XML language or that it should be translated into a standard canonical XML form. Neither the caching system described by Fradette nor the protocol broker taught by Graham deal with the special problems associated with caching XML requests.

The Examiner notes that "*Schroeder discloses an incoming data object in XML language that is translated into a standard canonical XML form.*" But the cited passage of Schroeder at paragraphs [0048] and [0049] does not deal with caching, but instead with "normalizing" received XML data objects (not requests) by passing them through standard XSL transforms. There is no suggestion in the cited passage of Schroeder that these "data objects" are requests or that, once "normalized" that they are stored or compared with previously stored prior requests. In short, there is nothing in the cited passage of Schroeder that suggests that incoming XML requests should be placed in canonical form so that they can be compared with previously stored canonical requests to identify prior requests that are logically identical to the incoming request as claimed. Nothing in Schroeder suggests that XML requests may be logically equivalent even though they have differing character content, and nothing in Schroeder suggests that this characteristic can lead to caching inefficiencies, and nothing in Schroeder suggests a solution to this problem. Like Mattis and Graham, Schroeder contains no hint of the claimed invention.

The Examiner stated that claims 6-9, 11 and 15-18 are rejected for similar reasons as stated with respect to claim 2. It is submitted that these claims are in fact allowable for the reasons stated above by applicants.

The rejection of claims 2, 6-9, 11, and 15-18 as being obvious in view of Fradette, Graham and Schroeder should be withdrawn for the reasons given above, as well as the reasons presented in connection with claims 1 and 3-5, above.

Conclusion

It is believed that this application is in condition for allowance.

Respectfully submitted,

Dated: March 16, 2008

Charles G. Call, Reg. 20,406

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this Response is being transmitted by facsimile to the central facsimile number of the U.S. Patent and Trademark Office, (571) 273-8300, on March 16, 2008.

Dated: March 16, 2008 Signature 

Charles G. Call, Reg. No. 20,406
USPTO Customer No. 021253
218 Seminole Court
Marco Island, FL 34145
Ph. (312) 265-0900 Fax (312) 277-3320
call@patentsoft.com